

# Adaptive Smoothing Techniques for 3-D Unstructured Meshes \*

Andrew Kuprat  
Los Alamos National Laboratory

## Abstract

To correctly capture the behaviour of deforming material volumes in 3-D, the Los Alamos unstructured grid code X3D has access to a variety of moving mesh algorithms. We present two such algorithms which markedly differ in their computational complexity. The first algorithm, Moving Finite Elements for Surfaces, has only “2-D” computational complexity, in that we only solve for interface motions and obtain volume point motions through interpolation. The second algorithm, Minimum Error Gradient Adaption, has “3-D” complexity, since the volume tetrahedral deformations must be computed. Naturally, the 3-D complexity algorithm can model realistically a larger class of physical problems than the lower complexity approach. We present examples in metallic grain growth and semiconductor process modeling.

**Key Words:** adaptive mesh smoothing, multimaterial grids, moving grids, moving surfaces, unstructured grids, moving finite elements.

## Two 3-D Smoothing Techniques with Differing Computational Complexity

In this paper we demonstrate two different smoothing techniques for 3-D unstructured meshes which differ in their computational complexity.

The first method, *Moving Finite Elements for Surfaces*<sup>1,2</sup>, moves the triangles of the interfaces between 3-D volumes composed of tetrahedra. Tetrahedral vertices that do not appear on an interface are moved by interpolation, while vertices appearing on interface triangles are moved using an implicit method. We may thus say that, although *volumes* are deformed by the moving grid, the computational complexity of the method is only “2-D”, not “3-D”.

This type of method is suitable for physical problems where it is acceptable to model material interfaces as discrete boundaries, with there being no need to accurately resolve the sharp boundary layers that actually constitute these interfaces. An example of this is movement of curved interfaces under mean curvature, as would occur in the growth of metallic grains.<sup>3</sup>

---

\* Work supported by the U.S. Department of Energy.

The second method, *Minimum Error Gradient Adaption* (MEGA), is suitable for more difficult physical problems where volume field quantities need to be defined at all points throughout the 3-D volume and/or detailed knowledge of boundary layers is important. Here we deform our volume tetrahedral elements in such a manner as to efficiently minimize errors in the piecewise linear representation of the volume field over the tetrahedra.

We give illustrations of both approaches which we have implemented in the Los Alamos multimaterial code X3D<sup>4,5</sup>.

### **“Moving Finite Elements for Surfaces” Applied to Metallic Grain Growth**

We use Moving Finite Elements for Surfaces to move a multiply-connected network of triangles for the modeling of deformation of 3-D grains. In metallic grain growth, interface surfaces obey the simple equation

$$v_n = K,$$

where  $v_n$  is the normal velocity of the interface, and  $K$  is the local mean curvature. We represent interfaces as parametrized surfaces:

$$\mathbf{u}(s_1, s_2) = \sum_{\text{nodes } j} \alpha_j(s_1, s_2) \mathbf{u}_j.$$

Here,  $(s_1, s_2)$  is the surface parametrization, the sum is over interface nodes  $j$ ,  $\alpha_j(s_1, s_2)$  is the piecewise linear basis function which is unity at node  $j$  and zero at all other interface nodes, and  $\mathbf{u}_j$  is the vector position of node  $j$ .

We have that

$$\dot{\mathbf{u}}(s_1, s_2) = \sum_j \alpha_j(s_1, s_2) \dot{\mathbf{u}}_j,$$

and

$$v_n = \dot{\mathbf{u}}(s_1, s_2) \cdot \hat{\mathbf{n}} \quad (\hat{\mathbf{n}} \text{ is local surface normal}).$$

So

$$v_n = \sum_j (\hat{\mathbf{n}} \alpha_j) \cdot \dot{\mathbf{u}}_j.$$

In effect, we have that the basis functions  $j$  for  $v_n$  are  $n_i\alpha_j$ , where  $\hat{\mathbf{n}} = (n_1, n_2, n_3)$ . These basis functions are discontinuous piecewise linear, since the  $n_i$  are piecewise constant.

Moving Finite Elements for Surfaces minimizes

$$\int (v_n - K)^2 dS$$

over all possible values for the derivatives  $\dot{\mathbf{u}}_j$ . (The integral is over the surface area of the interfaces.) We thus obtain

$$\begin{aligned} 0 &= \frac{1}{2} \frac{\partial}{\partial \dot{u}_j^i} \int (v_n - K)^2 dS, \quad i \in \{1, 2, 3\} \\ &= \int (v_n - K) n_i \alpha_j dS. \end{aligned}$$

This leads to a system of  $3N$  ODE's:

$$(n_i \alpha_j, n_k \alpha_l) \dot{u}_l^k = (K, n_i \alpha_j), \quad \text{or}$$

$$\mathbf{C}(\mathbf{y}) \dot{\mathbf{y}} = \mathbf{g}(\mathbf{y}),$$

where  $\mathbf{y}$  is the  $3N$ -vector containing the  $x$ ,  $y$ , and  $z$  coordinates of all  $N$  nodes,  $\mathbf{C}(\mathbf{y})$  is the matrix of inner products of basis functions, and  $\mathbf{g}(\mathbf{y})$  is the right-hand side of inner products involving surface curvature.

Although  $\mathbf{g}(\mathbf{y}) = (K, n_i \alpha_j)$  appears ill-defined for piecewise linear manifolds, being the inner product of a distribution  $(K)$  with discontinuous functions  $(n_i \alpha_j)$ , we can replace it by a well-defined sum of surface tensions over the triangular facets of the interfaces using an integral identity for manifolds.<sup>1</sup>

The advantage of this method is that our PDE solver need only loop over the interface triangles, and hence the complexity of the computation is “2-D”, even though 3-D volumes are deformed.

In Figures 1-4 we show a time sequence for deformation of four metallic grains which surround a fully-enclosed fifth grain. The central grain begins with a rough spherical shape (Fig. 1), changes into a smooth, curved tetrahedron under the action of surface tension (Figs. 2, 3), and disappears in finite time, leaving the four surrounding grains (Fig. 4). It should be noted that the outer surfaces of the surrounding grains are prevented from collapsing because of a Dirichlet boundary condition.

Due to the low 2-D computational complexity of the calculation, this run took only one-half hour on a workstation. For computation we used the Carlson/Miller 2-D Gradient Weighted Moving Finite Elements package (publicly available from [carlson@math.purdue.edu](mailto:carlson@math.purdue.edu)) which we have incorporated into our code X3D.

### **“MEGA” Applied to Oxidation/Diffusion of a Silicon Wafer**

If the physical problem being simulated involves tracking gradients of *volume* concentration fields or interface-defining boundary layers, any smoothing scheme must involve deformation of the volume elements, rather than just deformation of interface surface elements.

Minimum Error Gradient Adaption (MEGA) is a 3-D generalization of a 2-D adaptive mesh smoothing scheme by Bank and Smith<sup>6</sup>. In Minimum Error Gradient Adaption, we adjust the positions of the vertices so as to minimize the functional

$$F = \int_{\Omega} \|\nabla(u - u_L)\|^2 dx. \quad (1)$$

That is, the weighted  $\mathcal{L}_2$  norm of the gradient of the error between the true solution  $u(x, y, z)$  and its piecewise linear approximation  $u_L(x, y, z)$  on each tetrahedron.

Minimizing the gradient of the error leads to optimal resolution of solution gradients which can be crucial for correct calculation of diffusion profiles. A secondary benefit of minimizing the error gradient is that it works to prevent “tet collapse” as the mesh moves. This is because solution gradients are poorly represented on wafer-thin tetrahedra, and are thus avoided when minimizing this functional.

Since the exact solution  $u$  in (1) is generally unknown, the method is to approximate the error by the six quadratic “bump functions” associated with the edges of each tetrahedron. (The “bump” functions are the pairwise products of the four linear “hat” functions associated with the four vertices of each tetrahedron.) Thus in practice, all that is needed to evolve the mesh is an estimate of the error at each edge midpoint in the mesh. These are usually obtained as *a posteriori* error estimates computed when numerically solving differential equations.<sup>7</sup>

The example we give here involves the diffusion of boron in a semiconductor wafer, occurring simultaneously with deformation of the wafer by an oxidation process. For this example, we calculated time-dependent analytic diffusion profiles and boundary deformations that would mimic the results one would obtain if one actually solved the correct equations for boron diffusion and boundary motion. Using this analytic model, we computed error estimates at the edge midpoints in the mesh, and then we performed MEGA using these error estimates. Thus what we tested in this example was not the correctness of a PDE solver, but the feasibility of using the MEGA approach to adaptively smooth the tetrahedral mesh when good error estimates are available.

In Figures 5-8 we show a run with 9765 nodes in which the mesh is initially concentrated in a “tri-band” structure to minimize the gradient of the error of the boron concentration field. Not seen in this view, the concentration of the tetrahedral mesh by MEGA extends into the wafer and involves a concerted movement of tetrahedra throughout the volume.

In Figure 6, the oxidation front has deformed the upper surface of the wafer and MEGA has allowed the grid to “deconcentrate” due to the diffusion of the boron. Figures 7-8 show how the initially concentrated grid of  $< 10,000$  nodes produces better resolution of the boron field than a uniform hexahedral grid of 206,500 nodes.

## Conclusions

In the most general case of deforming volumes and volume concentration fields, one can obtain a large savings in time and space computational complexity by employing a

3-D mesh smoothing scheme such as MEGA. If the physical problem is simpler, in that local volume concentrations and interface-defining boundary layers need not be known with precision, then one can achieve an even greater savings in time by performing essentially a 2-D calculation using a method such as Moving Finite Elements for Surfaces. We accommodate both types of schemes in the versatile Los Alamos unstructured grid code X3D.

### References

1. K. Miller, *A Geometrical-Mechanical Interpretation of Gradient-Weighted Moving Finite Elements*, SIAM J. Num. Anal., to appear.
2. N. Carlson and K. Miller, *Gradient weighted moving finite elements in two dimensions*, in "Finite Elements Theory and Application", D.L. Dwyer, M.Y. Hussaini and R.G. Voight, editors, pp. 151-164, Springer Verlag, 1988.
3. D.A. Porter and K.E. Easterling, *Phase Transformations in Metals and Alloys*, Great Britain: Van Nostrand Reinhold, 1988, pp. 130-136.
4. D.C. George, X3D User's Manual, Los Alamos National Lab Report LA-UR-95-3608 (1995).
5. H.E. Trease, *Three-Dimensional Free Lagrangian Hydrodynamics*, Proceedings of the first Free-Lagrange Conference, Lecture Notes in Physics, Springer-Verlag, Vol. 238 (1985), pp. 145-157.
6. R.E. Bank and R.K. Smith, *Mesh smoothing using a posteriori error estimates*, SIAM J. Sci. Comp., to appear.
7. R.E. Bank and R.K. Smith, *Some a posteriori error estimates based on hierarchical bases*, SIAM J. Numerical Analysis, 30 (1993), pp. 921-935.



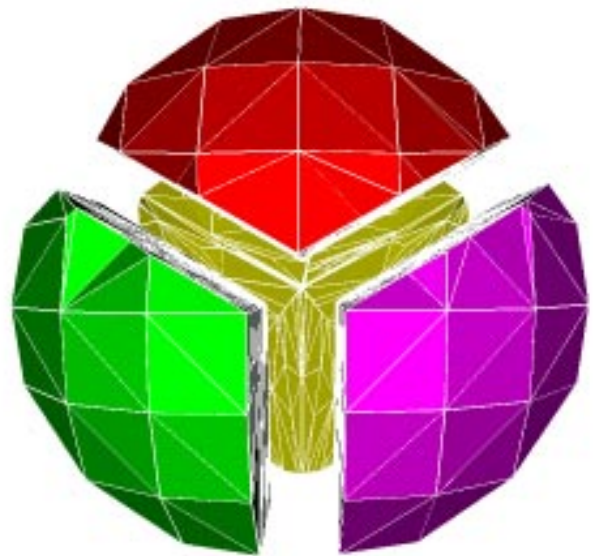
**Fig. 1:** Exploded view of initial configuration of 4 grains surrounding a central grain. Interfaces are jagged.



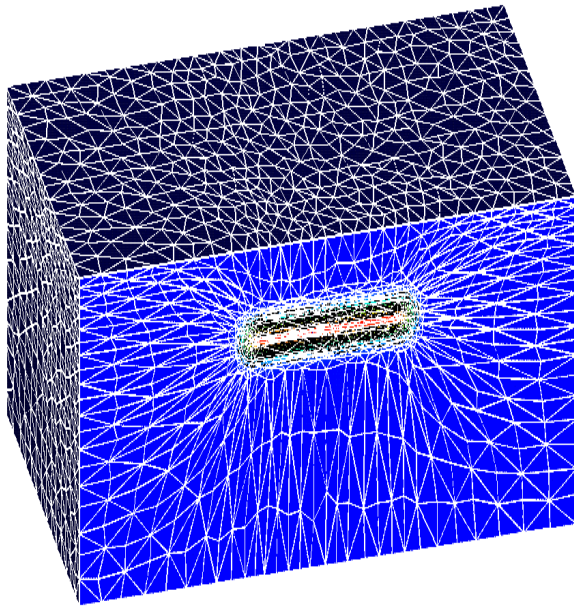
**Fig. 2:** Configuration at  $t = 0.1$  after evolution by surface tension (using Moving Finite Elements for Surfaces). Interfaces are smoothed; central grain has become a curved tetrahedron.



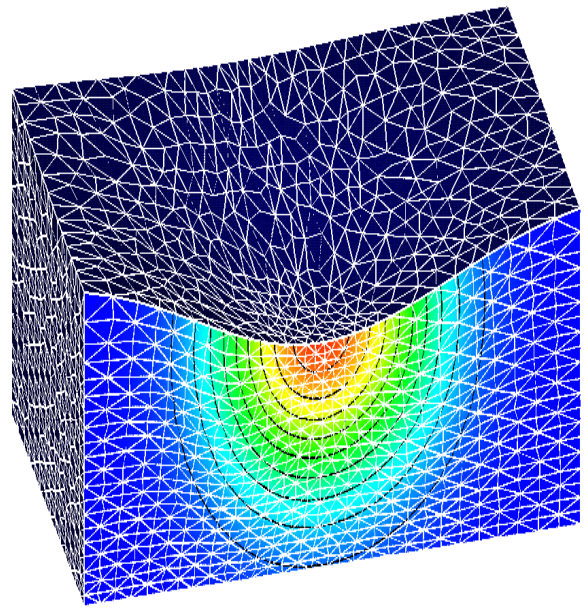
**Fig. 3:** Configuration at  $t = 0.4$ . Central grain is maintaining its shape, but shrinking.



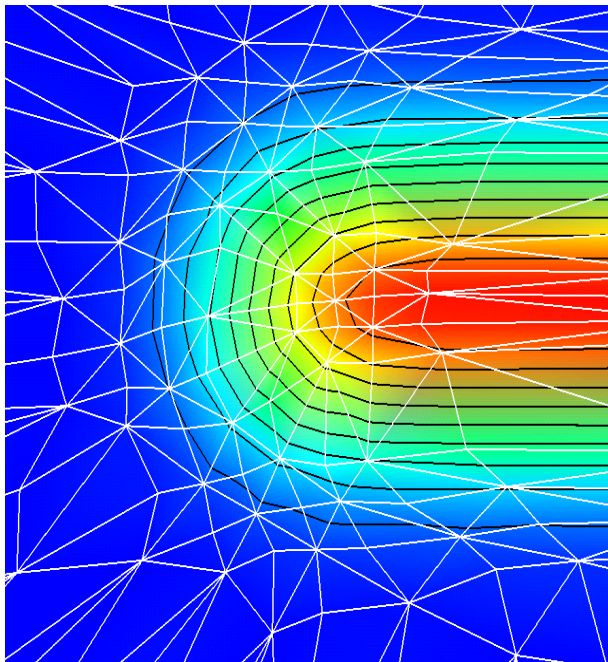
**Fig. 4:** Configuration at  $t = 0.6$ . Central grain has totally collapsed, leaving 4 grains with planar interfaces.



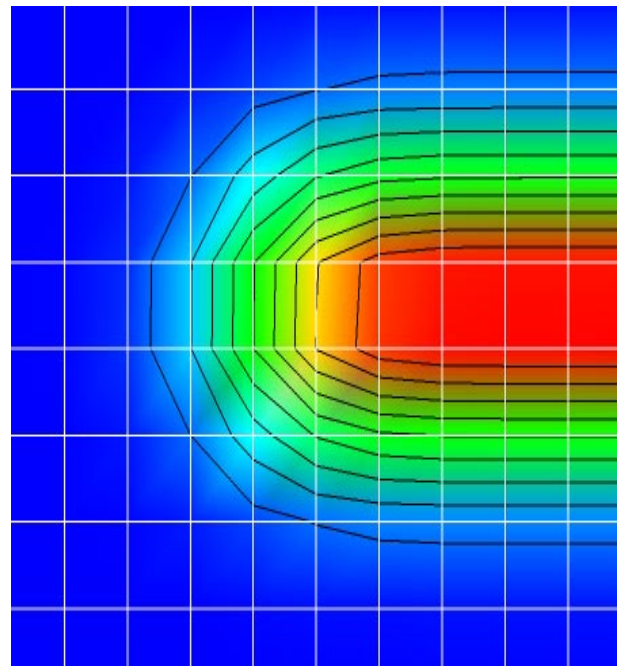
**Figure 5: MEGA smoothed grid with 9765 nodes at  $t=0$ .**



**Figure 6: Grid at  $t=8000$  showing boron contours.**



**Figure 7: MEGA smoothed grid with 9765 nodes at  $t=0$ . (Close-up showing boron contours.)**



**Figure 8: Same close-up as Fig. 7, but using 206,500 node uniform hexahedral grid.**